

REMARKS/ARGUMENTS

Claims 1-30 are pending in the application. Claims 1, 4-6, 8, 12, 14, 17-19, 21, 25, 27, and 30 have been amended. Reconsideration is respectfully requested. Applicants submit that the pending claims 1-30 are patentable over the art of record and allowance is respectfully requested of claims 1-30.

Applicants would like to thank Examiners Gaffin and Radtke for holding a telephone interview with their representative, Janaki K. Davda, on June 22, 2006, at 4:30 p.m. (EST).

The rejection of claim 14 under 35 U.S.C. 101 and proposed claim amendments made to claim 14 to overcome the rejection were discussed. In particular, Applicants representative pointed to Applicants' Specification, paragraph 93, describing that the "term 'article of manufacture' as used herein refers to code or logic implemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.) or a computer readable medium, such as magnetic storage medium (e.g., hard disk drives, floppy disks,, tape, etc.), optical storage (CD-ROMs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, firmware, programmable logic, etc.)" and submitted that hardware logic and a computer readable storage medium are tangible embodiments. No agreement was reached.

Proposed claim amendments to claim 1 and the cited Cox patent application were discussed. The Examiners submitted that the Cox patent application describes parallel parsing. Applicants respectfully traverse. Additionally, the Examiners submitted that a tuple is taught by attributes of the XML element being stored in a hashtable, as described on page 4, paragraph 57, of the Cox patent application. Applicants respectfully traverse. Applicants' tuples are records or rows that may be inserted into a table of a database (e.g., Specification, page 2, paragraph 4). No agreement was reached.

Proposed claim amendments to claim 8 and the cited Cox patent application were discussed. Applicants' representative pointed to page 2, paragraph 7, of Applicants' Specification, which describes a database load as loading data into a database in flat or delimited formats. Also, page 7, paragraph 19 of Applicants' Specification, describes that data is loaded

into a database serially, without requiring the generation of SQL commands. No agreement was reached.

Claims 12, 13, 25, and 26 are rejected under 35 U.S.C. 112, second paragraph.

Claims 12 and 25 recite the limitation "each processing unit", and the Examiner submits that there is insufficient antecedent basis for this limitation in the claim. Applicants respectfully traverse, however, to expedite prosecution, Applicants are amendment claims 12 and 25 to refer to "a processing unit".

Claims 13 and 26 recite "when restarting loading", and the Examiner submits that it is not clear to one of ordinary skill in the relevant art how the program can "restart loading" when there is no mention of "stopping loading". Applicants respectfully traverse. *MPEP Section 2106*, subheading *Claims Particularly Pointing Out and Distinctly Claiming the Invention*, recites that "the applicant need not explicitly recite in the claims every feature of the invention." Therefore, Applicants need not explicitly recite "stopping loading", and, Applicants' respectfully submit that it would be clear to one of ordinary skill in the relevant art that the phrase "when restarting loading" occurs after loading has stopped.

Claims 14 and 21 are rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter. Applicants respectfully traverse, however, to expedite prosecution, Applicants have amended claims 14 and 21. In particular, Applicants have amended claims 14 and 21 to recite an article of manufacture comprising one of hardware logic implementing logic and a computer readable storage medium. Again, Applicants' Specification, paragraph 93, describes that the "term 'article of manufacture' as used herein refers to code or logic implemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.) or a computer readable medium, such as magnetic storage medium (e.g., hard disk drives, floppy disks, tape, etc.), optical storage (CD-ROMs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, firmware, programmable logic, etc.)."

Claims 1-2, 5, 7-8, 12, 14-15, 18, 20-21, 25, and 27-28 are rejected under 35 U.S.C. 102(b) as being anticipated by Cox (U.S. Patent Application No. 2002/0112224 A1). Applicants respectfully traverse.

Anticipation requires that the identical invention must be shown in a single reference in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989).

Claim 1 describes generating a map specification that maps input data in the one or more input files to columns of tuples (e.g., Specification, page 8, paragraph 23 – page 10, paragraph 20; FIG. 3; Note that example (4) on page 10 illustrates two tuples that are produced from XML data in example (3)). Parallel processing of the one or more input files is performed to output tuples, wherein the parallel processing includes parallel parsing and construction of the tuples using the map specification (e.g., Specification, page 3, paragraph 10; page 6, paragraph 16; page 7, paragraphs 20-22; page 8, paragraph 23 – page 10, paragraph 20; page 11, paragraph 30; page 12, paragraph 33). The tuples are serially loaded into the data store while enforcing the order of the data in the one or more input files.

On the other hand, the Cox patent application describes a thread of the SAM XML parser is instantiated to process the recently received XML file into the XML elements (page 3, paragraph 42), and FIG. 4 illustrates a single SAX XML parser 42. The Cox patent application describes that, once an XML element is read from the XML file, all of the attributes of the element are stored in a hashtable in memory. The Cox patent application does not anticipate generating a map specification that maps input data in the one or more input files to columns of tuples and performing parallel processing of the one or more input files to output tuples, wherein the parallel processing includes parallel parsing and construction of the tuples using the map specification.

Additionally, claim 1 describes that tuples are serially loaded into the data store while enforcing the order of the data in the one or more input files. The Examiner submits that "the order remains intact because queue are First-In-First-Out data structures and the files are read from beginning to end. Applicants respectfully traverse. The Cox patent application describes generation of database commands, such as SQL statements, to execute against the database to load the data from the XML file into the database (page 1, paragraph 15) and multiple SQL processor threads that may execute the SQL commands to update the database. Because there

are multiple SQL processor threads executing the SQL commands, the order of the data in the one or more input files is not necessarily enforced.

Thus, claim 1 is not anticipated by the Cox patent application.

Claims 14 and 27 are not anticipated by the Cox patent application for at least the same reasons as were discussed with respect to claim 1.

Dependent claims 2, 5, 7, 14-15, 18, 20, and 28 incorporate the language of one of independent claims 1, 14, and 27 and add additional novel elements. Therefore, dependent claims 2, 5, 7, 14-15, 18, 20, and 28 are not anticipated by the Cox patent application for at least the same reasons as were discussed with respect to claims 1, 14, and 27.

Additionally, claim 5 describes that serially loading the tuples further comprises loading the tuples without generating SQL commands. The Specification, on page 2, paragraph 7, describes that a database loader typically allows data to be loaded into a database in flat or delimited formats. Page 7, paragraph 19 of Applicants' Specification, describes that data is loaded into a database serially, without requiring the generation of SQL commands. Applicants respectfully submit that database loaders are able to load data without requiring the generation of SQL commands.

Claim 8 and 21 describe, under control of a master row mapper, invoking one or more slave row mappers, wherein the slave row mappers perform processing in parallel with the master row mapper and with each other, wherein the parallel processing includes parallel parsing and construction of tuples using a map specification that maps input data in the one or more input files to columns of the tuples (e.g., Specification, page 8, paragraph 23 – page 10, paragraph 20; FIG. 3; Note that example (4) on page 10 illustrates two tuples that are produced from XML data in example (3)), processing data in a first input file to output tuples, and forwarding the tuples and tuples in one or more spillfiles to one or more database loader processes (e.g., Specification, page 6, paragraph 16). Claims 8 and 21 also describe, under control of each of the slave row mappers, processing data in a separate input file to output tuples and storing the tuples in a corresponding spillfile (e.g., Specification, page 6, paragraph 16). Claims 8 and 21 further describe, under control of the one or more database loader processes, serially loading the tuples into the data store (e.g., Specification, page 7, paragraph 19).

The Examiner submits that the "master row mapper" is read on "Loader". Applicants respectfully traverse. In the Cox patent application, the Loader merely parses the XML file

(paragraph 52), but the Loader does not invoke one or more slave row mappers, wherein the slave row mappers perform processing in parallel with the master row mapper and with each other, wherein the parallel processing includes parallel parsing and construction of tuples using a map specification that maps input data in the one or more input files to columns of the tuples, process data in a first input file to output tuples, and forward the tuples and tuples in one or more spillfiles to one or more database loader processes.

The Examiner submits that "slave row mapper" is read on "Operator". Applicants respectfully traverse. In the Cox patent application, an Operator generates the appropriate database command statements (paragraph 53), but the Operator does not process data in a separate input file to output tuples and store the tuples in a corresponding spillfile. Additionally, unlike the claimed slave row mappers, the Operator does not perform processing in parallel with the master row mapper and with each other, wherein the parallel processing includes parallel parsing and construction of tuples using a map specification that maps input data in the one or more input files to columns of the tuples.

Additionally, the Cox patent application does not anticipate under control of the one or more database loader processes, serially loading the tuples into the data store. Instead, the Cox patent application describes that the Operator generates the appropriate database command statements, preferable SQL commands, which are executed to update the database (paragraphs 56, 46, and 53).

Thus, claims 8 and 21 are not anticipated by the Cox patent application.

Dependent claims 12 and 25 incorporate the language of independent claims 8 and 21, respectively, and add additional novel elements. Therefore, dependent claims 12 and 25 are not anticipated by the Cox patent application for at least the same reasons as were discussed with respect to claims 8 and 21.

Claims 6 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cox in view of Reeder (U.S. Patent No. 5,852,812). Applicants respectfully traverse.

Claims 6 and 19 describe that the tuples output when processing data from each of the input files are appended to a separate temporary storage location and, when serial loading is interrupted, restarting the serial loading using the tuples in the separate temporary storage locations without reprocessing the one or more input files.

The Reeder patent does not cure the defects of the Cox patent application with reference to independent claims 1 and 14, from which claims 6 and 19 depend. For example, the Reeder patent does not teach or suggest generating a map specification that maps input data in the one or more input files to columns of tuples, performing parallel processing of the one or more input files to output tuples, wherein the parallel processing includes parallel parsing and construction of the tuples using the map specification, and serially loading the tuples into the data store while enforcing the order of the data in the one or more input files.

Additionally, the Reeder patent describes that if the import does not complete without errors, then all new records that were placed in the import database are deleted, and the records are re-imported into the database after examination (Col. 14, line 65-Col. 15, line 2). On the other hand, claims 6 and 19 restart the serial loading using the tuples in the separate temporary storage locations *without reprocessing* the one or more input files. These input files are in hierarchical format (from independent claims 1 and 14). The Reeder patent's examination and re-importing of data teaches away from restarting the serial loading using the tuples in the separate temporary storage locations without reprocessing the one or more input files.

Thus, claims 6 and 19 are not taught or suggested by the Cox patent application or the Reeder patent, either alone or in combination.

Claims 3, 9-11, 16, 22-24, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cox in view of Kloos et al. (U.S. Patent No. 6,556,999). Applicants respectfully traverse.

The Kloos patent does not cure the defects of the Cox patent application with reference to independent claims 1, 14, and 27 from which claims 3, 16, and 29 depend. For example, the Kloos patent does not teach or suggest generating a map specification that maps input data in the one or more input files to columns of tuples, performing parallel processing of the one or more input files to output tuples, wherein the parallel processing includes parallel parsing and construction of the tuples using the map specification, and serially loading the tuples into the data store while enforcing the order of the data in the one or more input files.

Also, claims 3, 16, and 29 describe, while performing processing of a first section from the multiple sections under control of a first row mapper, determining that there has been an error in logically dividing the physical input file, continuing processing of a next section from

the multiple sections that is also being processed by a second row mapper, and notifying the second row mapper to terminate processing of the next section.

The Examiner submits that the Cox patent application teaches performing processing of a first section from the multiple sections under control of a first row mapper and logically dividing the physical input file. Applicants respectfully traverse. The Cox patent application describes that once an XML element is read from the XML file, all of the attributes of the element are stored in a hashtable in memory (paragraph 57), but there is no teaching or suggestion of logically dividing the physical input file.

The Kloos patent describes that when an error is detected with the data retrieved from RDE product or during loading of that data into O/C database, the front-end site from which the data came will be marked as "bad" causing further loads of data from that front-end site to be disabled until the problem is corrected (Col. 40, lines 54-63). The Kloos patent also describes that once the problem is resolved, the front-end site is re-enabled (Col. 40, lines 54-63). The Examiner submits that "in processing" is read on "with the data retrieved from the RDE product". Applicants respectfully submit that this phrase is not used in claims 3, 16, and 29. Moreover, the Kloos patent does not teach or suggest the claimed first row mapper and second row mapper and the claimed interactions. The Kloos patent only refers to a front-end site. Also, the Kloos patent merely describes that an error has been detected, but this does not teach or suggest determining that there has been an error in logically dividing the physical input file.

Thus, claims 3, 16, and 29 are not taught or suggested by the Cox patent application or the Kloos patent, either alone or in combination.

The Kloos patent does not cure the defects of the Cox patent application with reference to independent claims 8 and 21 from which claims 9-11 and 22-24 depend. For example, the Kloos patent does not teach or suggest, under control of a master row mapper, invoking one or more slave row mappers, wherein the slave row mappers perform processing in parallel with the master row mapper and with each other, wherein the parallel processing includes parallel parsing and construction of tuples using a map specification that maps input data in the one or more input files to columns of the tuples, processing data in a first input file to output tuples, and forwarding the tuples and tuples in one or more spillfiles to one or more database loader processes; under control of each of the slave row mappers, processing data in a separate input

file to output tuples and storing the tuples in a corresponding spillfile; and, under control of the one or more database loader processes, serially loading the tuples into the data store.

Claims 9 and 22 describe, under control of the master row mapper, determining that there has been an error in processing the data in at least one input file and terminating the slave row mappers. The Examiner submits that "in processing" is read on "with the data retrieved from the RDE product". Applicants respectfully submit that parallel processing is described as including parallel parsing and construction of tuples using a map specification that maps input data in the one or more input files to columns of the tuples in independent claims 8 and 21, from which claims 9 and 22 depend, and such processing is not taught or suggested by the Kloos patent. Moreover, neither the Cox patent application nor the Kloos patent, either alone or together, describes the claimed slave row mappers, which perform processing in parallel with the master row mapper and with each other, wherein the parallel processing includes parallel parsing and construction of tuples using a map specification that maps input data in the one or more input files to columns of the tuples, each of which processes data in a separate input file to output tuples and stores the tuples in a corresponding spillfile.

Claims 10 and 23 describe, under control of the master row mapper, determining that there has been an error in loading the processed data in at least one input file and terminating the slave row mappers. The Kloos patent describes disabling a front-end site, but neither the Cox patent application nor the Kloos patent, either alone or together, describes the claimed master row mapper and slave row mappers.

Claims 11 and 24 describe, under control of at least one of the slave row mappers, determining that there has been an error in processing the data in at least one input file and terminating each of the other slave row mappers processing a separate input file whose order follows the separate input file being processed by the slave row mapper that determined that there has been an error. The Examiner submits that "in processing" is read on "with the data retrieved from the RDE product". Applicants respectfully submit that parallel processing is described as including parallel parsing and construction of tuples using a map specification that maps input data in the one or more input files to columns of the tuples in independent claims 8 and 21, from which claims 11 and 24 depend, and such processing is not taught or suggested by the Kloos patent. Moreover, but neither the Cox patent application nor the Kloos patent, either alone or together, describes the claimed slave row mappers.

Thus, claims 9-11 and 22-24 are not taught or suggested by the Cox patent application or the Kloos patent, either alone or in combination.

Claims 4, 17, and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cox in view of Kloos et al. and further in view of Holenstein et al. (U.S. Patent No. 7,003,531). Applicants respectfully traverse.

The Holenstein patent does not cure the defects of the Cox patent application and Kloos patent with reference to independent claims 1, 14, and 27 from which claims 4, 17, and 30 depend. For example, the Holenstein patent does not teach or suggest performing parallel processing of one or more input files to output data, wherein the parallel processing includes parallel parsing and construction of tuples using the map specification, and serially loading the tuples into the data store while enforcing the order of the data in the one or more input files.

Claims 4, 17, and 30 describe that the tuples output when processing data from each of the input files are appended to a separate temporary storage location and deleting the temporary storage location into which the second row mapper was appending the tuples from the processing of the next section.

The Examiner submits that "temporary storage location" is read on "hashtable". Applicants respectfully traverse. The temporary storage locations store tuples, while the hashtable stores attributes of an XML element (paragraph 57). The Holenstein patent describes that if any rows are missing from the block of data but are present in the target database, then such rows are deleted (Col. 12, lines 55-57). Deleting rows from the target database does not teach or suggest deleting the temporary storage location into which the second row mapper was appending the tuples from the processing of the next section.

Thus, claims 4, 17, and 30 are not taught or suggested by the Cox patent application, the Kloos patent, or the Holenstein patent, either alone or in combination.

Claims 13 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cox in view of Holenstein et al. Applicants respectfully traverse.

The Holenstein patent does not cure the defects of the Cox patent application with reference to independent claims 8 and 21 from which claims 13 and 26 depend. For example, the Holenstein patent does not teach or suggest, under control of a master row mapper, invoking

one or more slave row mappers, wherein the slave row mappers perform processing in parallel with the master row mapper and with each other, wherein the parallel processing includes parallel parsing and construction of tuples using a map specification that maps input data in the one or more input files to columns of the tuples, processing data in a first input file to output tuples, and forwarding the tuples and tuples in one or more spillfiles to one or more database loader processes; under control of each of the slave row mappers, processing data in a separate input file to output tuples and storing the tuples in a corresponding spillfile; and, under control of the one or more database loader processes, serially loading the tuples into the data store.

Thus, claims 8 and 21 are not taught or suggested by the Cox patent application or Holenstein patent, either alone or together. Claims 13 and 26 are not taught or suggested by the Cox patent application or Holenstein patent, either alone or together, at least by their dependency on claims 8 and 21.

Conclusion

For all the above reasons, Applicants submit that the pending claims 1-30 are patentable over the art of record. Applicants have not added any claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 09-0460.

The attorney of record invites the Examiner to contact her at (310) 553-7973 if the Examiner believes such contact would advance the prosecution of the case.

Dated: June 23, 2006

By: _____/Janaki K. Davda/_____

Janaki K. Davda
Registration No. 40,684

Please direct all correspondences to:

David Victor
Konrad Raynes & Victor, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984